

Docket No. 40101-03901

booting instructions through a network. (Godse, col 5, lines 25-30.) The policy file cited by the Examiner is described as "a file comprising a set of entries that control the booting process." (Godse, col. 6, lines 24-25.) The purpose of the policy file is to specify if software or data entities are to be acquired from the remote system manager or from a local memory unit. (Godse, col. 6, lines 32-35.) Before each stage of the booting process, the policy file is examined to determine if the boot-up stage, *e.g.* discovery, software download, software initialization and data fill, must be run locally or remotely. (Godse, col. 6, lines 36-38.)

As described in Godse, the policy file contains the preferred order for the stages of the booting process to occur and the preferred location (local or remote) from which the stage should proceed. Thus, while the policy file has control over the order and location of the booting process, it does not have any functionality which allows the computer to recover specific data. For example, the policy file will determine when the software download stage occurs in the booting process and the location from which the software will be downloaded. If the software download stage is to be carried out locally, the process described with respect to the flow chart of Fig. 7 will be used. (Godse, col. 8, line 27 - col. 9, line 51.) If the local software download is successful, the policy file indicates the next boot stage which should be carried out, *e.g.*, software initialization. (Godse, col. 9, lines 24-27) If the local software load is unsuccessful or if the policy file indicates the software download is to be carried out remotely, the software download is attempted via the remote system. (Godse, col. 9, lines 42-51.) The remote software download process is described with respect to the flow chart of

Docket No. 40101-03901

Fig. 8. (Godse, col. 9, line 51 - col. 10, line 14.) If the software download stage is not completed remotely, it is aborted (step 806 in Fig. 8). (Godse, col. 9, lines 64-66.) Thus, the policy file will not allow the boot process to be completed because the next stage (*e.g.*, software initialization) will not begin until the previous stage (*e.g.*, software download) is complete. Each of the stages of the boot process are described as operating in the same manner.

In contrast to Godse, claim 1 of the present invention recites:

- setting an archive-commit flag on;
- committing said archive to said persistent storage;
- setting said archive commit flag off;
- setting a list-commit flag on;
- committing a list of said required objects to said persistent storage
- setting said list-commit flag off;

As described in the specification of the present application as an exemplary embodiment of according to the present invention, if the device experiences a power failure or shut down during the archive committing step the archive-commit flag will remain in the on state. Upon subsequent initialization of the device, the gateway G will determine whether the archive-commit flag is on and query the repository manager as to whether the archive was successfully committed to storage. If the committing operation was not successful, the device will re-attempt the file transfer. (Specification, page 8, lines 8-16). Thus, the use of the archive-commit flag allows the recovery of a specific data element or object (*e.g.*, the archive) that was lost due to the failure of the device.

Docket No. 40101-03901

Likewise, if the device experiences a power failure or shut down during the list committing step the list-commit flag will remain in the on state. Upon subsequent initialization of the device, the gateway G will determine the list-commit flag is on and determine that the list committing step has failed, thereby commencing a recovery by instructing the registry to examine the contents of the repository to ascertain the list of objects that should be present in the registry. The list committing step may then be completed. (Specification, page 8, line 26 - page 9, line 3). Thus, the use of the list-commit flag allows the recovery of the specific data (*e.g.*, the list of objects) that was lost due to the failure of the device.

In contrast, the policy file of Godse does not allow the computer to recover specific data that was lost during the failure. A computer operating according to Godse, regardless of when the failure occurred, will re-initialize from the start of the boot up sequence as defined in the policy file. The boot up sequence will be started from the beginning with all the subsequent stages being carried out. Indeed, the Examiner stated in the Office Action that neither the setting of the archive commit flag nor the setting of the list commit flag was taught by Godse.

Furthermore, Godse does not suggest the present invention as recited in claim 1. There is no indication in Godse that one of ordinary skill in the art would be motivated by the policy file described in Godse in order to recover specific data that is lost during power failure or shutdown, much less employ the method recited by claim 1. For at least these

Docket No. 40101-03901

reasons, Applicants respectfully submit that claim 1 is not rendered obvious by Godse, and is allowable. Claims 2-5 and 7-8 depend directly or indirectly from claim 1, and at least for that reason are also allowable.

The Rejection of Claims 6, 9, 10 and 11 Should Be Withdrawn

The Applicants respectfully request that the Examiner reconsider the rejection of these claims based on the following.

Britt describes that the status of a flag indicating loss of power is checked prior to writing any block of downloaded data into flash memory 22b. (Britt, col. 12, lines 5-13.) If the flag indicates that power has been interrupted, the writing routine ends. If the flag indicates power is still available, the routine continues writing the next block of data into flash memory. (Britt, col. 12, lines 13-18.)

A "NUM_BLOCKS" field is provided in flash memory to indicate how many blocks have been written prior to the power being lost. (Britt, col. 12, lines 18-20.) Once power is restored and a connection to the network is reestablished, the downloading resumes. Since the number of blocks already written is recorded in the flash memory, only the data blocks not already written in memory need to be downloaded, and the download resumes from the last block that was successfully written to flash memory 22b.

If the power is lost during the writing of the data block, a sustaining device (e.g., a battery, charged capacitor, etc.) ensures that writing of the current block can be completed.

Docket No. 40101-03901

(Britt, col. 12, lines 20-23.) Thus, the device described in Britt cannot result in a partially written data block because the sustaining device allows the device to completely finish writing the current data block if the primary power is lost.

In contrast to Britt, claim 6 recites “instructing said persistent storage to clear the portion of said archive committed to persistent storage when said step of determining establishes that an archive was being committed when said device was powered-off.” For example, if there is a power failure during the step of committing the archive to persistent memory, only a portion of the archive may have been committed prior to the power loss. Thus, with respect to the data itself (the archive), claim 6 recites that upon initialization the portion of the archive that was stored before the power failure is cleared.

The Examiner maintains in the Office Action that Britt inherently teaches this clearing of the portion of the archive, stating that “the number of blocks of data written before power off is maintained so that if a block of data was not completely written the associated block number field is not maintained and thus if the [sustaining device] was not present there inherently would be an erasure step taken of the part of the data block being written into memory in which a failure in recording the block number occurred.”

The Applicants respectfully disagree with the Examiner’s characterization of the teaching of Britt. As an initial matter, inherency applies only to features necessarily present, not features that might be present if modification were made to the disclosure. MPEP § 2112 states:

Docket No. 40101-03901

The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. To establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.

M.P.E.P. § 2112 (citations omitted).

The Examiner's assertion that modification of the system of Britt will result in the present invention as recited in claim 6 is contrary to law. However, even if one were to remove the sustaining device, Britt does not teach or suggest clearing the portion of the archive. For example, if the device described in Britt has a portion of a data block written into memory prior to a power failure, upon subsequent initialization, the device may begin downloading with the partially written block based on the block number field described in Britt. However, Britt neither teaches (specifically or inherently) nor suggests that the portion of the data block that was previously written would be cleared.

Moreover, to the extent that the Examiner's rejection is more properly a § 103 rejection, the Applicants respectfully submit that Britt teaches away from the present invention because it specifically teaches the use of the sustaining device to avoid partially storing blocks altogether. There is no discussion within Britt of any cases where a sustaining device is not used and consequently a partially written data block is not possible within the

Docket No. 40101-03901

device of Britt. Thus, the concept of clearing a partially written data block is not contemplated by that reference.

Claim 6 specifically recites the clearing of the portion of the archive committed before the power failure. Applicants respectfully submit that Britt does not anticipate claim 6 because it neither teaches nor suggests this clearing of the portion of the archive that was written to memory before the power failure. Accordingly, claim 6 is allowable. Claims 9-11 depend directly or indirectly from claim 6, and at least for that reason are also allowable.

Information Disclosure Statement filed July 20, 2001

Applicants filed an Information Disclosure Statement ("IDS") and an accompanying Form PTO-1449 on July 20, 2001. Applicants are attaching a copy of the IDS and the accompanying Form PTO-1449 including the certificate of mailing to evidence the filing of these documents and copies of the listed references. For the Examiner's convenience, copies of the non-patent publications disclosed in the IDS are included. The Examiner is invited to request additional copies of the disclosed patents from the Applicants if such copies were lost.

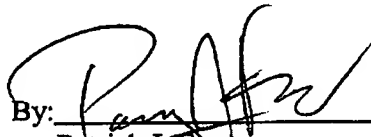
Docket No. 40101-03901

CONCLUSION

In view of the amendment and remarks submitted above, the Applicants respectfully submit that the present case is in condition for allowance. All issues raised by the Examiner have been addressed, and a favorable action on the merits is thus earnestly requested.

Respectfully submitted,

Dated: June 14, 2002

By: 
Patrick J. Fay
Reg. No. 35,308

FAY KAPLUN & MARCIN, LLP
150 Broadway, Suite 702 Floor
New York, NY 10038
(212) 619-6000 (phone)
(212) 208-6819 (facsimile)